What is claimed is:

1.   1.   A method of performing a transaction in a database system,
2.   comprising:
3.   receiving a transaction to be performed, wherein the transaction is
4.   processed by a plurality of access modules; and
5.   performing a flush of a transaction log in each access module
6.   before an end transaction procedure.

1.   2.   The method of claim 1, further comprising issuing a request to
2.   flush the transaction log with a message sent to each access module for
3.   performing a last step of the transaction.

1.   3.   The method of claim 2, further comprising avoiding performance of
2.   a transaction log flush in the end transaction procedure.

1.   4.   The method of claim 2, further comprising determining that the last
2.   step is being performed by all of the plurality of access modules.

1.   5.   The method of claim 1, further comprising determining if the
2.   transaction log has been flushed before performing the end transaction
3.   procedure.

1.   6.   The method of claim 5, further comprising avoiding performance of
2.   a transaction log flush in the end transaction procedure if the transaction log has
3.   been flushed.

1.   7.   The method of claim 1, further comprising:
2.   identifying the transaction as an implicit transaction.

8. The method of claim 1, further comprising:

performing the end transaction procedure, which follows execution of the transaction.

9. The method of claim 8, performing the end transaction procedure comprising:

skipping broadcast of a directive indicating commencement of the end transaction procedure to the plurality of access modules.

10. A method of performing an end transaction procedure in a database system, comprising:

a first access module in the database system writing an end transaction indication to a first transaction log portion, the first access module being part of a cluster of access modules; and

the first access module sending an end transaction directive to a fallback module associated with the first access module, the fallback module being part of the cluster.

11. The method of claim 10, wherein the first access module sends the end transaction directive to the fallback module but not to other access modules in the cluster.

12. The method of claim 10, wherein sending the end transaction directive comprises sending an end transaction-part one directive.

13. The method of claim 12, further comprising the first access module broadcasting an end transaction-part two directive to all access modules in the cluster.

1   14.   The method of claim 10, further comprising the fallback module

2 writing an end transaction indication to a second transaction log portion.

1   15.   The method of claim 10, further comprising the first access module

2 flushing the first transaction log portion.

1   16.   The method of claim 10, further comprising the first access module

2 flushing the first transaction log portions but the other access modules in the

3 cluster not flushing their respective transaction log portions.

1   17.   A database system comprising:

2   a plurality of storage media; and

3   a plurality of access modules, wherein each access module is

4 coupled to one of the plurality of storage media; and

5   each of the access modules being adapted to flush a transaction log

6 before performing an end transaction procedure.

1   18.   The database system of claim 17, further comprising a controller

2 adapted to determine if each access module has flushed the transaction log

3 maintained by the access module.

1   19.   The database system of claim 18, wherein the controller is adapted

2 to skip sending a directive to perform a transaction log flush if the controller

3 determines that each access module has flushed the transaction log before the

4 end transaction procedure.

1   20.   The database system of claim 17, further comprising a controller

2 adapted to provide a flush directive with a message to each of the access

3 modules to perform a last step of the transaction.

1 21. An article comprising a medium storing instructions for enabling a
2 processor-based system to:
3   receive a transaction to be performed, wherein the transaction is
4 processed by a plurality of access modules ;
5   determine that a last step of the transaction involves the plurality of
6 access modules; and
7   flush a transaction log to a storage while the last step is performed
8 by the plurality of access modules.

1 22. The article of claim 21, further storing instructions for enabling the
2 processor-based system to:
3   perform an end transaction, wherein the end transaction follows
4 execution of the transaction.

1 23. The article of claim 22, further storing instructions for enabling a
2 processor-based system to:
3   avoid broadcast of a directive indicating commencement of the end
4 transaction to the plurality of access modules.

1 24. A method of performing a transaction in a database system,
2 comprising:
3   receiving a transaction to be performed on plural access modules in
4 the database system;
5   maintaining a log to track operations performed in the transaction;
6   writing the log to persistent storage before start of an end
7 transaction procedure.

1 25. The method of claim 24, wherein writing the log to persistent
2 storage comprises flushing the log.

1 26. The method of claim 24, wherein maintaining the log comprises
2 maintaining a transaction log.

1 27. The method of claim 24, further comprising performing the end
2 transaction procedure, the end transaction procedure comprising writing an end
3 transaction indication into the log.

1 28. A database system comprising:
2   storage media;
3   access modules coupled to the storage media; and
4   a parsing engine coupled to the access modules, the parsing engine
5 adapted to perform one of:
6   (a)   providing a directive with a message to perform a last
7 step of a transaction and communicating the directive to the access modules,
8 each access module responsive to the directive to perform a transaction log flush
9 before performance of an end transaction procedure; and
10   (b)   determining if each of the access modules has
11 performed a transaction log flush before start of the end transaction procedure;
12   the parsing engine adapted to avoid sending a broadcast directive
13 to the access modules to cause performance of a transaction log flush during the
14 end transaction procedure.